

# hwscan3: TCC processor inventory report



John W. Shipman

2009-08-26 15:57

## Abstract

Describes a system for presenting information on the hardware configuration of our servers and clients.

This publication is available in Web form<sup>1</sup> and also as a PDF document<sup>2</sup>. Please forward any comments to [tcc-doc@nmt.edu](mailto:tcc-doc@nmt.edu).

## Table of Contents

1. Requirements .....	1
2. Presentation .....	1
3. Sources of information .....	2
3.1. Information from LDAP .....	2
3.2. Information from the GetHardware XML-RPC service .....	2
3.3. The scanreport.xml file: Specifying report parameters .....	2
4. Table design principles .....	5
5. Information to be presented .....	5
6. Web layout .....	6
6.1. Format of the server table .....	6
6.2. Room page layout .....	7

## 1. Requirements

---

New Mexico Tech Computer Center (TCC) users need a centralized report of TCC workstations and user-accessible servers. For example, a user might need to know processor speeds and memory capacities in order to schedule a long-running process, or to find systems with a specific peripheral device.

Internal documentation for this project, including all code in literate programming style, can be found in the *hwscan3 IMS*<sup>3</sup>.

## 2. Presentation

---

The intended audience for this report is the community of TCC users. The project does not address the needs of the TCC Engineering and Maintenance groups for special reports.

Given the nearly universal access our user base has to the Web, the obvious choice for the presentation of TCC's hardware inventory is as a set of Web pages that are regenerated periodically by a **cron** job.

<sup>1</sup> <http://www.nmt.edu/tcc/hw/hwscan3/>

<sup>2</sup> <http://www.nmt.edu/tcc/hw/hwscan3/hwscan3.pdf>

<sup>3</sup> <http://www.nmt.edu/tcc/hw/hwscan3/ims3/>

The initial implementation will inventory only processors and selected peripherals on those processors. Later revisions may add printers and other standalone online resources.

## 3. Sources of information

---

Information required for this report comes from three places:

- Section 3.1, “Information from LDAP” (p. 2).
- Section 3.2, “Information from the GetHardware XML-RPC service” (p. 2).
- Section 3.3, “The scanreport.xml file: Specifying report parameters” (p. 2).

### 3.1. Information from LDAP

For an overview of the TCC LDAP directory, see *LDAP attribute cross-reference*<sup>4</sup>.

The LDAP tree for TCC client systems (including login servers) is:

```
ou=clients,ou=machines,dc=tcc,dc=nmt,dc=edu
```

The entries in this tree have class `tccHardware`. Attributes of interest from these objects are:

**cn**

The client name, e.g., “speare4a-2-2”.

**tccOfficeMachine**

TRUE if the machine is in someone's office and not publicly accessible.

**tccSpecialOS**

Undefined for dual-boot machines. Permissible values are “Linux” or “Mac OS X”.

### 3.2. Information from the GetHardware XML-RPC service

Most of the information for this report comes from an XML-RPC service described in the TCC Wiki under `Software/Local/Forge/ForgeWeb/GetHardware`<sup>5</sup>.

### 3.3. The scanreport.xml file: Specifying report parameters

In order to generate the specified report, we need certain pieces of information over and above the set of *SysInfo* reports:

- A way to differentiate classroom workstations from open lab workstations.
- A way to differentiate Linux-only systems from dual-boot workstations.
- A list of servers that users need to know about. Most servers (e.g., **ldap0**) are nothing users need to access directly.
- Which device types (e.g., **scanner**) should appear in the device report.

---

<sup>4</sup> <http://www.nmt.edu/tcc/doc/ihs/ldap/>

<sup>5</sup> <https://fedora.nmt.edu/tccwiki/Software/Local/Forge/ForgeWeb/GetHardware>

## Note

Ideally, these bits of information should live in our LDAP directory. However, for the moment, we'll need to maintain a small data file. Keeping it in XML allows us to use standard tools, such as *nxml-emacs*<sup>6</sup>, to maintain the file in a valid state.

Here's the schema, `scanreport.rnc`, expressed in Relax NG<sup>7</sup> format. The root element is called **scanreport**:

```
scanreport.rnc
start = scanreport
scanreport = element scanreport
{
  roomlist,
  systemlist,
  devicelist
}
```

The **roomlist** element defines the set of room prefixes, differentiating them as type **class** for classrooms and **open** for open labs.

- The **type** is “c” for classrooms, “o” for open labs.
- The **prefix** attribute is used to match rooms against client names. For example, the names of all the clients in MSEC-III room 187 building start with “msec187-”. In the attribute here, we'll omit the trailing hyphen.
- The **full** attribute is the formal name of the room, e.g., “MSEC-III 187”.

```
scanreport.rnc
roomlist = element roomlist
{
  room*
}
room = element room
{
  attribute type { 'c' | 'o' },
  attribute prefix { text },
  attribute full { text },
  empty
}
```

The **systemlist** element is a list of systems. Each **system** element names one system that is either a public server, a Linux-only box, or both.

```
scanreport.rnc
systemlist = element systemlist { system* }
system = element system
{
  attribute nodename { text },
  attribute server { '0' | '1' }?,
  attribute linux-only { '0' | '1' }?,
  empty
}
```

The **devicelist** element enumerates which device types are of interest to users. The order of the included **device** elements dictates the order of their presentation in the device report. The **kind** attribute

<sup>6</sup> <http://www.nmt.edu/tcc/help/pubs/nxml>

<sup>7</sup> <http://www.nmt.edu/tcc/help/pubs/rnc/>

must be uppercase, since its values must correspond to device information returned by the `GetHardware()` interface.

scanreport.rnc

```
devicelist = element devicelist
{ device*
}
device = element device
{ attribute kind { text },
  empty
}
```

Here is the entire `scanreport.xml` file. Typing a **make** command in the directory containing this document will generate the file.

scanreport.xml

```
<!--Do not edit this file directly. The original is
! in file hwscan3.xml, and 'make' will rebuild it.
!-->
<scanreport>
  <roomlist>
    <room type='c' prefix='cramer101' full='Cramer 101'/>
    <room type='c' prefix='cramer120' full='Cramer 120'/>
    <room type='o' prefix='fidel130' full='Fidel 130'/>
    <room type='c' prefix='jonesa101' full='Jones A101'/>
    <room type='c' prefix='jonesa104' full='Jones A104'/>
    <room type='o' prefix='library122' full='Library 122'/>
    <room type='c' prefix='library207' full='Library 207'/>
    <room type='c' prefix='library208' full='Library 208'/>
    <room type='c' prefix='msec101' full='MSEC-III 101'/>
    <room type='c' prefix='msec103' full='MSEC-III 103'/>
    <room type='c' prefix='msec105' full='MSEC-III 105'/>
    <room type='c' prefix='msec187' full='MSEC-III 187'/>
    <room type='c' prefix='speare4' full='Speare 4'/>
    <room type='c' prefix='speare4a' full='Speare 4A'/>
    <room type='c' prefix='speare4b' full='Speare 4B'/>
    <room type='o' prefix='speare5' full='Speare 5'/>
    <room type='o' prefix='speare15' full='Speare 15'/>
    <room type='c' prefix='speare23' full='Speare 23'/>
    <room type='c' prefix='speare23a' full='Speare 23A'/>
    <room type='c' prefix='speare23b' full='Speare 23B'/>
    <room type='c' prefix='speare102' full='Speare 102'/>
    <room type='c' prefix='speare113' full='Speare 113'/>
    <room type='c' prefix='speare116' full='Speare 116'/>
    <room type='o' prefix='speare142' full='Speare 142'/>
    <room type='c' prefix='weir102' full='Weir 102'/>
    <room type='c' prefix='weir128' full='Weir 128'/>
    <room type='c' prefix='weir209' full='Weir 209'/>
    <room type='c' prefix='workman101' full='Workman 101'/>
  </roomlist>
  <systemlist>
    <system nodename='speare5-1-9' linux-only='1'/> <!--pi-->
    <system nodename='speare5-1-14' linux-only='1'/> <!--corvette-->
    <system nodename='speare5-1-15' linux-only='1'/> <!--cobra-->
    <system nodename='speare5-1-16' linux-only='1'/> <!--boardwalk-->
```

```
<system nodename='speare5-1-17' linux-only='1' /> <!--eldorado-->
<system nodename='speare142-1-5' linux-only='1' /> <!--underdog-->
<system nodename='login' server='1' />
<system nodename='login64' server='1' />
<system nodename='rainbow' server='1' />
</systemlist>
<devicelist>
  <device kind='VIDEO' />
  <device kind='MONITOR' />
  <device kind='OPTICAL' />
  <device kind='SCANNER' />
</devicelist>
</scanreport>
```

## 4. Table design principles

---

The bulk of the output will be in the form of tables where each row describes one client. Here are some general design principles for Web-based tables:

- Columns should be as narrow as possible so as to present the maximum information without requiring noxious horizontal scrolling. However, they should not be too narrow if it obscures the meaning.
- Column heads must also be narrow, but not too narrow to make sense.
- More important columns should be further to the left, in case the table doesn't all fit on the screen.
- If a column may become large, it should move further to the right, especially if it's not one of the more important fields.

## 5. Information to be presented

---

Before we consider presentation, let's summarize all the different configuration information that users might need to know.

- Server name, e.g., "speare116-2-1".
- Number of CPUs. Currently, this is either 1 or 2.
- Processor speed in MHz.
- Architecture, typically "i686".
- Memory size in MB.
- Operating system: dual-boot, Linux-only, or MacOS.
- Video controllers. Some systems have more than one.
- Monitor size, vendor, and type, if known.
- Optical drives.
- Scanners, if any.

## 6. Web layout

---

The start page for the generated report will reside at <http://www.nmt.edu/tcc/hw/index.html>. We'll try to keep this page fairly short, moving bulky sections to separate pages.

There are three main categories of machines to be displayed:

- Linux servers for remote login. This category includes only servers in the server room, not Linux-only clients, so it's a pretty short list.
- Open labs. At this writing we have only five lab rooms.
- Classrooms. Currently we maintain around twenty classrooms.

To keep the start page from getting too large, it will be structured like this.

- A table labeled "Linux servers for remote login." See Section 6.1, "Format of the server table" (p. 6).
- A bullet list entitled "Open labs". Within this list, each bullet is a link to a *room page* containing a table of the clients in that lab. For the layout of this table, see Section 6.2, "Room page layout" (p. 7). Room page names will be the same as the room prefix; for example, the room page for Speare 5 will be named "speare5.html".
- A table entitled "Classrooms". On the start page, we'll display a table with one row per classroom. Within this table, the room name will be a link to the room page displaying the clients in that room. For the format of the room page, see Section 6.2, "Room page layout" (p. 7).

It's important to distinguish between two types of classroom: those with an instructor system only, and those that have a workstation at every seat. Consequently, rather than presenting the classrooms as a simple bullet list with one bullet per room, we will display a table with one row for each room, and these columns:

- Name of the room, e.g., "cramer101".

We derive the room name from the machine name by discarding everything from the client name starting at the first hyphen, if any. So, for example, client `speare5-1-11` is considered to be in room `speare5`.

- Number of workstations (other than the instructor machine).
- Number of instructor machines—0 or 1. This will almost always be 1, so we display this information in the last column.

There is one case where the count of instructor machines will be zero for a classroom. Speare 4 is currently divided into three "rooms." The instructor machine is `speare4-inst`, but the names of all the client machines in that room start with either "speare4a" or "speare4b". Thus, room `speare4` has one instructor machine and zero clients, while rooms `speare4a` and `speare4b` have zero instructor machines and many clients.

### 6.1. Format of the server table

Because these systems are in the server room and not user-accessible, we won't need to present information on peripherals. Here are the columns in this table, in order.

- Server name, e.g., "login".
- Number of CPUs.
- Processor speed in MHz.
- Architecture, typically "i686".
- Memory size in MB.

## 6.2. Room page layout

For each open lab or classroom, there will be a room page displaying the information on the clients in that room as a table.

- Client name, e.g., `speare5-1-6`.
- Number of CPUs.
- Processor speed in MHz.
- Architecture, e.g., `"i686"`.
- Memory size in MB.
- Operating system: one of `"Linux/Windows"`, `"Linux-only"`, or `"Mac OS X"`.
- List of peripherals of interest to users. Because there are several types, we won't put each type in a separate column; that would make the table much too wide.

Instead, there will be one column labeled `"Peripherals"`, with vertically stacked blocks for each peripheral.

Each block will start with a label such as `"Video:"`, `"Monitor:"`, `"Optical:"`, or `"Scanner:"`, followed by the description of that peripheral.

These blocks will use a hanging indent, so that if the entry for one peripheral does not fit within the width of a cell, continuation lines will be indented further.

